Assembled Genomic Scaffolds

Files in this archive

List of file names and number of scaffolds in parentheses:

- Agri.genomic.scaffs.fasta (74,159)
- Agri.genomic.scaffs.RM.fasta (74,159)
- Agri.scaffs.to.LGs.csv (3,278)

File Format

Agri.genomic.scaffs.fasta:

```
>[SCAFFOLD_ID] [LENGTH] [COVERAGE]
SEQUENCE
```

where `[SCAFFOLD_ID]` is the scaffold identification number assigned to the scaffold by ABySS, `[LENGTH]` is the length of the scaffold, and `[COVERAGE]` is the total k-mer coverage. The sequence headers are the ABySS-generated scaffold names. For more information, check the ABYSS manual.

Agri.genomic.scaffs.RM.fasta:

```
>Agri_[SCAFFOLD_ID]
SEQUENCE
```

where `[SCAFFOLD_ID]` is the scaffold identification number assigned to the scaffold by ABySS, and the `SEQUENCE` is the hard-masked scaffold sequence. *NOTE*: `Agri.genomic.scaffs.RM.fasta` is the hard-masker version of `Agri.genomic.scaffs.fasta` (i.e., all repetitive regions have been replaces with `N`'s')

Agri.scaffs.to.LGs.csv:

The file are in comma-separated values (CSV) format with two columns:

1. the ABySS-assigned scaffold ID (scaff)
2. linkage group (LG)

Genome Annotation

Files in this archive

- Agri.gene.models.only.gff (15,848 genes)
- Agri.gene.models.evidence.bombyx_func.gff (2,657,992 lines)
- Agri.predicted.proteins.bombyx_func.fasta (15,848 sequences)
- Agri.predicted.transcripts.bombyx_func.fasta (15,848 sequences)

File Format

Agri.gene.models.only.gff:

This file is a standard GFF format as generated by MAKER2 accessory scripts.

Agri.gene.models.evidence.bombyx_func.gff:

This file is a standard GFF format as generated by MAKER2 accessory scripts. In addition, functional annotation information based on *Bombyx mori* proteins from UniProt is included in the 'Note' field.

Agri.predicted.transcripts.bombyx_func.fasta & *Agri.predicted.proteins.bombyx_func.fasta*:

The two files are in FASTA format. The sequence names were generated by MAKER2.

Phenotypes and Consensus, Linkage Group-specific Markers

Files in this archive

List of file names and the table dimensions in (rows x columns) format:

- Agri.FL-BC.consensus.geno.matrix.csv (468 x 38)
- Agri.KS-BC.consensus.geno.matrix.csv (450 x 38)
- Agri.KS-SG.consensus.geno.matrix.csv (201 x 38)

These consensus genotype matrices were generated by collapsing all genotyping data for each individual for each linkage group to a single, consensus genotype. The (original) genotyping data are under `genotype_matrices` (for details, see "Materials and Methods").

File Format

The files are in R/qtl format: the columns contain phenotypes and marker information, and the rows contain linkage group (LG), marker position, and individual information.

ROWS

The first 3 rows in each file contain similar information:

- Row 1: phenotype names and marker names
- Row 2: empty in phenotype columns, and LG information in the columns containing the markers (M1-M30, one for each LG)
- Row 3: empty in phenotype columns, and marker position for each marker (in this case, 0)

Additionally,

- In the FL-BC file, rows 4 through 468 correspond to the 465 recombinant individuals in the FL-BC population.
- In the RK-BC file, rows 4 through 450 correspond to the 447 recombinant individuals in the KS-BC population.
- In the KS-SG file, rows 4 through 201 correspond to the 198 recombinant individuals in the KS-SG population.

COLUMNS

All files contain 8 phenotype columns:

1. individual ID (ind)
2. family ID (family)
3. collection date (year)
4. developmental time (devTime)
5. weight at eclosion (weight)
6. pulse-pair rate (pr)
7. peak amplitude (pa)
8. asynchrony interval (ai)
9. The FL-BC file contains an additional 30 columns corresponding to a single consensus marker for each of the 30 LGs, with M1 being homologous to the Z chromosome.
10. The KS-BC file contains an additional 30 columns corresponding to a single consensus marker for each of the 30 LGs, with M1 being homologous to the Z chromosome.
11. The KS-BC file contains an additional 30 columns corresponding to a single consensus marker for each of the 30 LGs, with M1 being homologous to the Z chromosome.

Assembled Transcripts

Files in this archive

List of file names and number of sequences in parentheses:

- Agri.transcripts.fasta (96,420)

File Format

Agri.transcripts.fasta:

```
>[TRANSCRIPT_ID]
SEQUENCE
```

where `[TRANSCRIPT_ID]` is the transcript identification assigned to the sequence by Trinity. For more information, check the Trinity output user guide.

Marker Sequences and Marker-to-Linkage Group Tables

1. Marker Sequence Files

List of file names along with the number of sequences in parentheses:

- Agri.FL-BC.marker.sequences.fasta (5,721)
- Agri.KS-BC.marker.sequences.fasta (8,091)
- Agri.KS-SG.marker.sequences.fasta (12,801)

File Format

The files are in FASTA format:

```
>[POPULATION]_[MARKER]
SEQUENCE
```

where `[POPULATION]` corresponds to one of FL-BC, KS-BC, KS-SG, and `[MARKER]` corresponds to the marker ID.

2. Markers-to-LG Tables

List of file names along with the number of sequences in parentheses:

- Agri.FL-BC.marker2LG.csv (5,721)
- Agri.KS-BC.marker2LG.csv (8,091)
- Agri.KS-SG.marker2LG.csv (12,801)

File Format

The tables contain information linking marker IDs to linkage groups in 3 columns:

1. marker ID (marker)

2. linkage group (LG)
3. position along the LG (position)

Genotype Matrices (genotypes for all markers associated with linkage groups)

Files in this archive

List of file names and the table dimensions in (rows x columns) format:

- Agri.FL-BC.genotype.matrix.csv (468 x 5,729)
- Agri.KS-BC.genotype.matrix.csv (450 x 8,099)
- Agri.KS-SG.genotype.matrix.csv (201 x 12,809)

File Format

The files are in R/qtl format: the columns contain phenotypes and marker information, and the rows contain linkage group (LG), marker position, and individual information.

ROWS

The first 3 rows in each file contain similar information:

- Row 1: phenotype names and marker names
- Row 2: empty in phenotype columns, and LG information in the columns containing the markers
- Row 3: empty in phenotype columns, and marker position along the LG in the columns containing the markers
    - *NOTE*: in the KS-SG file, the marker positions are simple placeholders since the markers in the segregant population could not be ordered.

Additionally,

- In the FL-BC file, rows 4 through 468 correspond to the 465 recombinant individuals in the FL-BC population.
- In the RK-BC file, rows 4 through 450 correspond to the 447 recombinant individuals in the KS-BC population.

- In the KS-SG file, rows 4 through 201 correspond to the 198 recombinant individuals in the KS-SG population.

COLUMNS

All files contain 8 phenotype columns:

1. individual ID (ind)
2. family ID (family)
3. collection date (year)
4. developmental time (devTime)
5. weight at eclosion (weight)
6. pulse-pair rate (pr)
7. peak amplitude (pa)
8. asynchrony interval (ai)

Additionally,

- The FL-BC file contains 5,721 columns corresponding to as many genetic markers across 30 LGs.
- The KS-BC file contains 8,091 columns corresponding to as many genetic markers across 30 LGs.
- The KS-BC file contains 8,091 columns corresponding to as many genetic markers across 30 LGs.

GENOTYPES

In the FL-BC file, the following genotypes appear:

- *aa*: homozygous for *FL* allele (*FL*/*FL*)
- *ab*: heterozygous (*FL*/*KS*)

In the KS-BC file, the following genotypes appear:

- *aa*: homozygous for *KS* allele (*KS*/*KS*)
- *ab*: heterozygous (*KS*/*FL*)

In the KS-SG file, the following genotypes appear:

- *aa*: homozygous for *KS* allele (*KS*/*KS*)
- *ab*: heterozygous (*KS*/*FL*)

Generate Marker Catalog (interrogate.cstacks.catalog.2.py)

Interpreter: Python 2

Command Line Use

```
python interrogate.cstacks.catalog.2.py PREFIX_A PREFIX_B
```

Inputs

- `PREFIX_A` : Prefix for first `catalog.tags.tsv` (generated by Stacks (V1) `cstacks`)
- `PREFIX_B` : Prefix for second `catalog.tags.tsv` (generated by Stacks (V1) `cstacks`)

Outputs

- `matches.txt` : tab-delimited file; rows correspond to a catalog entry; two columns: individual 1 stack_ID, individual 2 stack_ID

- `catalog.stats.txt` : a text file containing the number of stacks in individual 2 that correspond to any stack in individual 1, and vice versa

Reformat Genotype File for LepMAP2 (format.stacks.output.for.lepmap.py)

Interpreter: Python 2

Command Line Use

```
python format.stacks.output.for.lepmap.py PREFIX
```

Inputs

- `hybrids.haplotypes.tsv` : haplotypes file for parent1 (generated by Stacks (V1) `genotypes`)
- `PREFIX.genotypes.tsv` : genotypes file for population (generated by Stacks (V1) `genotypes`)
- `PREFIX.haplotypes.tsv` : haplotypes file for population (generated by Stacks (V1) `genotypes`)

Outputs

- `PREFIX.lepmap.linkage` : input file for LepMAP2 in LINKAGE format
- `PREFIX.lepmap.markernames.txt` : every line contains the marker name for every marker included in the `PREFIX.lepmap.linkage` file in preserved order
- `PREFIX.perMarkerStats.txt` : for each marker, number of aa, ab, bb, and NA genotypes called; tab-delimited
- `PREFIX.perIndividualStats.txt` : for each individual, number of aa, ab, bb, and NA genotypes called; tab-delimited

Connect LepMAP2 Markers to Stack IDs (get.marker.positions.py)

Interpreter: Python 2

Command Line Use

```
python get.marker.positions.py PREFIX
```

Inputs

- `PREFIX.lepmap.markernames.txt` : marker names for the markers used to generate the map (generated by `format.stacks.output.for.lepmap.py`)
- `PREFIX.lod20.lg*X*.rmdup1.order.txt` : genetic map files - one for each linkage group (generated by LepMAP2)
- `rk.rf.lgs.match.txt` : each line contains information about the correspondence between a linkage group from each backcross population as well as the number of overlapping markers; tab-delimited

Outputs

- `PREFIX.genetic.map.with.stacks.names.csv` : each line corresponds to a marker in the genetic map with 4 elements: stacks_ID, linkage group, position, LepMAP2 marker name; tab-delimited

- `PREFIX.markers.txt` : every line contains the marker name for every marker included in the `PREFIX.lepmap.linkage` file in preserved order

Extract Marker Sequences (extract.marker.sequences.from.catalog.py)

Interpreter: Python 2

Command Line Use

```
python extract.marker.sequences.from.catalog.py PREFIX
```

Inputs

- `batch_0.catalog.tags.tsv`: catalog of markers (generated by Stacks (V1) `cstacks`)
- `PREFIX.genetic.map.with.stacks.names.csv`: each line corresponds to a marker in the genetic map with 4 elements: stacks_ID, linkage group, position, LepMAP2 marker name; tab-delimited (generated by `get.marker.positions.py`)

Outputs

- `PREFIX.marker.sequences.fasta`: the genomic sequences of the markers in the genetic map in FASTA format

Get Z-linked *B. mori* Proteins (grab.bombyx.z.proteins.py)

Interpreter: Python 3

Command Line Use

```
python grab.bombyx.z.proteins.py
```

Inputs

- `correspondence_table_Bmscaf_nscaf.txt` : AGP file for the *B. mori* genome
- `Bombyx_mori.GCA_000151625.1.32.gff3` : Unpacked GFF annotation file distributed with the *B. mori* genome (ftp://ftp.ensemblgenomes.org/pub/release-32/metazoa/gff3/bombyx_mori/)
- `Bombyx_mori.GCA_000151625.1.pep.all.fa` : Unpacked protein database distributed with the *B. mori* genome (ftp://ftp.ensemblgenomes.org/pub/release-32/metazoa/fasta/bombyx_mori/pep/)

Outputs

- `bm.zchrom.proteins.fa` : the amino acid sequences for Z chromosome proteins in FASTA format

Get Z-linked *M. cinxia* Proteins (get.melitaea.protein.IDs.py)

Interpreter: Python 3

Command Line Use

```
python get.melitaea.protein.IDs.py
```

Inputs

- `melitaea_scaff_ids.txt` : contains the scaffold IDs associated with the Z chromosome; every scaffold ID is on a separate line
- `Melitaea_cinxia_v1.gff` : GFF annotation file distributed with the *M. cinxia* genome
- `Melitaea_cinxia_proteins_v1.fa` : Protein database distributed with the *M. cinxia* genome

Outputs

- `mc.zchrom.proteins.fa` : the amino acid sequences for Z chromosome proteins in FASTA format
- `mc.zChrom.protein.IDs.txt` : the names of the proteins extracted, one protein name per line